Two Dimensional Cross-Layer Optimization for Packet Transmission over Fading Channel

Xiaofeng Bai, *Student Member, IEEE,* and Abdallah Shami, *Member, IEEE*

***Abstract*—In this paper a single-input-single-output wireless data transmission system with adaptive modulation and coding over correlated fading channel is considered, where run-time power adjustment is not available. Higher layer data packets are enqueued into a finite size buffer space before being released into the time-varying wireless channel. Without fixing the physical layer error probability, the objective is to minimize the average joint packet loss rate due to both erroneous transmission and buffer overflow. Two optimization techniques are incorporated to achieve the best solution. The first is policy domain optimization that formulates the data rate adaptation design as classical Markov decision problem. The second is channel domain optimization that appropriately partitions the channel variation based on particular fading environment and carried traffic pattern. The derived policy domain analytical model can precisely map any policy design into various QoS performance metrics with finite buffer setup. We then propose a tractable suboptimization framework to produce different two-dimensional suboptimal solutions with scalable complexity-optimality tradeoff for practical implementations.**

***Index Terms*—Finite state Markov channel, adaptive modulation and coding, Markov decision process, performance optimization, suboptimal scheduling.**

I. Introduction

With rapid development of wireless broadband applications, recently, cross-layer protocol design has emerged to be a practical yet necessary research subject. The main challenges of MAC-PHY cross-layer design for wireless communications are twofold. At the physical layer, on one side, multi-path propagation leads to a wireless fading channel over which signals are distorted and attenuated randomly before reception. At the MAC layer, on the other side, the limited buffer space is significantly vacated or exhausted from time to time, due to bursty arrivals from upper layers. These facts impose severe obstacles for providing *quality-of-service* (QoS) to upper layer applications. In this paper, we address the cross-layer optimization issues by jointly considering adaptive transmission scheduling and wireless fading channel partitioning, to achieve better performance gain. The major contributions of this work are summarized as follows: 1) This work proposes an advanced algorithm for *adaptive modulation and coding* (AMC)-enabled multirate transmission scheduling, where run-time power adjustment is not available. 2) This work develops an analytical model of the proposed algorithm for evaluating various QoS performance metrics. 3) Combined with previous research, this work proposes a new method for cross-layer performance optimization. 4) Based on the optimal solution, this work introduces a tractable design framework that generates a set of suboptimal schemes with scalable complexity-optimality tradeoff, for practical implementation.

The rest of this paper is organized as follows. In Section II we briefly review some related research work. In Section III we give the system model used for this study. The multirate transmission scheduling is formulated as a Markov decision problem in Section IV and further analyzed in Section V. In Section VI and VII the two-dimensional optimization and suboptimization design frameworks are exposited, respectively. Finally, we present possible extensions of this work in Section VIII before conclude in Section IX.

Manuscript received April 25, 2007; revised August 29, 2007 and October 18, 2007; accepted October 20, 2007. The associate editor coordinating the review of this paper and approving it for publication was E. Hossain. This work was supported in part by NSERC Discovery grant 262375-07 and NSERC Postgraduate scholarship 333629-2006. A preliminary version of this paper appeared in the Proceedings of IEEE ICCCN 2007.

X. Bai and A. Shami are with the Department of Electrical and Computer Engineering, The University of Western Ontario, London, ON, Canada, N6A 5B9 (e-mail: xbai6@uwo.ca; ashami@eng.uwo.ca).

Digital Object Identifier 10.1109/T-WC.2008.070378

II. Related Research

MAC-PHY cross-layer performance optimization for wireless communications has been studied by many researchers [1]–[11]. Most notably, the *finite state Markov channel* (FSMC) modeling and *adaptive modulation and coding* are commonly accepted as the fundamental techniques for developing effective cross-layer protocols and algorithms. The principle of FSMC is to discretize the continuous channel fading process into finite number of states, over which the channel can be qualified separately and utilized in different ways. AMC, coupled with FSMC, enables various adaptive techniques whereby the data transmission rate and/or power is dynamically tuned-up according to instantaneous *channel state information* (CSI). Many research works have presented different *channel partition methods* (CPMs) for the FSMC model [3], [12], [13]. For example, in [3] the authors proposed a CPM that maintains a certain level of average *packet error rate* (PER) over the time-varying channel when corresponding AMC mode is applied for each channel state. This work reported that varying the target PER that defines the FSMC model may lead to significant variation in the system performance. Hence, an effective way to minimize the overall packet loss, i.e., both at PHY and MAC layers, is to find the optimal value of target PER through cross-layer analysis. Along with various CPMs defining the FSMC model, the research on AMC-enabled adaptive transmissions includes the average power/delay(dropping) tradeoff [4], [6], or the minimization of average power/delay with constraint on the other [5], [7], by varying the instantaneous transmission power and data rate. The general principle of power/delay optimization is to, based

1536-1276/08$25.00 © 2008 IEEE

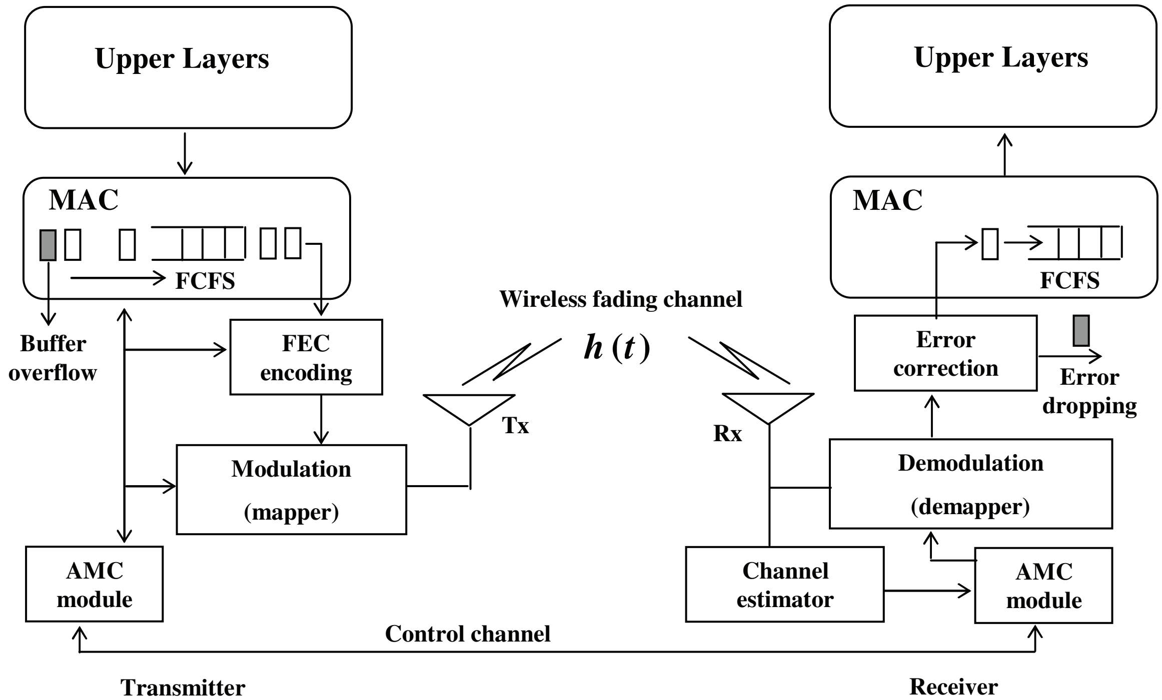

Fig. 1. Illustration of wireless packet transmission system model.

on some information theory, convert the channel error rate as function of particular channel state into minimum required power for the same channel state, that theoretically achieves a fixed (e.g., *mutual information* model in [4]) or bounded (e.g., *M-QAM transmission* model in [5]) channel error rate. The remaining two *heterogeneous* design objectives (power and delay) are related with a negotiable factor (e.g., β in [4]–[6]) indicating their relative importance, or related with one objective being constrained, to define the combined optimization target. Many of these researches investigate the application of *dynamic programming* (DP) and *Markov decision process* (MDP) theories for the optimal transmission policy design.

In some practical scenarios, slot-by-slot power adjustment may not be available, e.g., due to the inter-user interference in multiuser multichannel applications as discussed in [14]. Having the various power/delay oriented research, in this study, we intend to improve Liu's solution [3] by applying intelligent transmission scheduling. Without online power adjustment, in Liu's model the channel error is not fixed or bounded. In this case, there are two *homogeneous* design objectives (channel error and buffer overflow) that jointly define the optimization target, i.e., end-to-end packet loss. To avoid manipulating both sides of packet loss, in this study we convert the optimization target into the MAC layer throughput for the optimal transmission policy design.

III. System Model

Fig. 1 illustrates the general structure of the system model used in this study. At the transmitter side, packet stream arriving from upper layers are enqueued into a limited MAC buffer space. Backlogged packets are dequeued, on first-come-first-serve basis, and further processed by the *forward error correction* (FEC) encoder and digital modulator before being transmitted. Undergoing distortion and attenuation by the wireless fading channel, received symbols are sequentially passed through demodulator and decoder, with uncorrectable codewords being dropped, to retrieve MAC layer packets. At the PHY layer, AMC is applied to achieve adaptive multirate transmission over the time-varying fading channel.

FSMC is a well accepted block fading channel model for slow-varying flat fading channels, where the channel is assumed to stay in the same state within one block period. In this study we model the Rayleigh fading channel as FSMC and define a block as a time frame duration, which contains multiple codeword transmissions. Specifically, let $0 = \Gamma_0 < \Gamma_1 < \Gamma_2 < \cdots < \Gamma_{K-1} < \Gamma_K = \infty$ be the *signal-to-noise ratio* (SNR) thresholds partitioning the channel status, the channel is said to be in state $k = 1, 2, \cdots K$, if the received SNR falls into the interval of $[\Gamma_{k-1}, \Gamma_k)$. It is known that the received SNR of Rayleigh fading channel is characterized by the *probability density function* (PDF) of [15]: $p_r(r) = \frac{1}{\rho} e^{-\frac{r}{\rho}},\ r \geq 0$, where ρ denotes the average received SNR per symbol. In such a slow-varying channel model, the channel states are assumed only transit between neighboring states [3], [5], [8], [9], [12]. The corresponding channel state transition probabilities are given in [12].

IV. Problem Formulation

We apply five transmission modes as convolutionally coded *M-ary* QAM (*quadrature-amplitude modulation*), i.e., TM2 defined in [3], to represent the AMC-enabled multirate transmission, and refer them to as mode 1 to mode 5, respectively. Moreover, there is a *null* mode, referred to as mode 0, where no packet transmission takes place due to deteriorated channel condition or empty buffer. The rate adaptation functionality is accomplished through the collaboration of AMC modules located at the transmitter and receiver. As in [3]–[10], we assume an instant error-free control channel between the two AMC modules. Therefore, the CSI and selected transmission mode index are known by both sides of the transmission. This assumption can be partially released in practice by applying a memory space that temporarily buffers received modulation symbols before the demodulator, which allows the selected transmission mode index to be correctly received through possible retransmissions. Concurrently, incorrectly delivered CSI, though usually with strong FEC encoding and robust modulation, only causes instantaneously misapplied transmission mode and does not directly introduce transmission errors. More theoretical research with delayed/erroneous CSI is discussed in [11], [16]. We define *decision epoch* as $\mathbf{T} := \{0, T_f, 2T_f, \cdots, \infty\}$, with T_f denoting the duration of one time frame. The transmission mode is decided by the transmitter at each decision epoch and remains unchanged within this time frame. Furthermore, packets arriving at time $t \in (nT_f, nT_f+T_f]$ are enqueued at decision epoch nT_f+T_f and can be serviced in one of the subsequent time frames. Dropped packets by overflowed buffer are not serviced at later time.

A. Research Motivation

Our research is motivated by the following thoughts: 1) Both of the CPM forming the FSMC model and the selection of transmission mode effectively affect the system performance, and both are under the full discretion of the designer. 2) For a certain CPM, the transmitter should select particular transmission mode at each decision epoch based on instantaneous PHY layer channel state as well as MAC layer queuing information, such that the long-term probability of packet loss, i.e., either due to buffer overflow or erroneous transmission, is minimized. 3) The optimal solution for the system design is achieved by applying 2) over the most appropriate CPM, in terms of minimizing long-term end-to-end packet loss.

For a particular CPM, in order to select the best transmission mode, the transmitter side AMC module should consider: 1) the instantaneous channel state; 2) the number of packets available for transmission in the MAC buffer; 3) the number of packets in the MAC buffer at the next decision epoch, if particular transmission mode is selected. Therefore, the AMC module is making its optimal decision, on frame-by-frame basis, to meet a long-run target. This sequential decision-making problem can be optimized by the classical MDP theory.

B. MDP Based Optimization

The principle of MDP-based optimization is to determine the optimal policy[1] for the MDP such that the resultant Markov process offers the optimal performance of interest. If the resultant process by applying a certain policy is completely ergodic, the average per-transition reward, i.e., the process gain, is independent from the initial state and is only determined by the applied policy [18]. In this case, the value of the process gain, denoted as g, is given in [18, ch. 2] as

$$g = \boldsymbol{\pi} \times \mathbf{q}^T \tag{1}$$

where $\boldsymbol{\pi}$ is the steady state probability distribution vector of the resultant Markov process and $\mathbf{q}$ is the corresponding expected immediate reward vector, where the sth element is defined as $q_s = \sum_{\omega \in \mathcal{S}} (p_{s;\omega} \times r_{s;\omega})$, with $p_{s;\omega}$ and $r_{s;\omega}$ being the transition probability from state s to state ω within the state space $\mathcal{S}$, and the one-step reward function associated with such a state transition, respectively.

Proposition 1: The value of g in (1) driven by a given policy μ can be alternatively calculated from the corresponding state transition probability matrix $\mathbf{P}^\mu$ and expected immediate reward vector $\mathbf{q}^\mu$ as:

$$g = \mathbf{1}_{1\times N} \times [\mathbf{I}_N - \mathbf{P}^\mu + \mathbf{1}_{N\times N}]^{-1} \times [\mathbf{q}^\mu]^T \tag{2}$$

where $\mathbf{1}_{m\times n}$ is a $m \times n$ matrix with all elements are 1 and N is the total number of states in $\mathcal{S}$.

Proof: For a given policy μ, in steady state there exists $\boldsymbol{\pi}^\mu \times \mathbf{P}^\mu = \boldsymbol{\pi}^\mu$ and $\sum_{i\in\mathcal{S}} \pi_i^\mu = 1$. To include the latter unity property into the former balance equation, these two expressions can be augmented into matrix form as:

$$\boldsymbol{\pi}^\mu \times \mathbf{P}^\mu = \boldsymbol{\pi}^\mu, \qquad \boldsymbol{\pi}^\mu \times \mathbf{1}_{N\times N} = \mathbf{1}_{1\times N} \tag{3}$$

Noting that any full-ranked square matrix is nonsingular, the difference of the two expressions in (3) leads to

$$\boldsymbol{\pi}^\mu = \mathbf{1}_{1\times N} \times [\mathbf{I}_N - \mathbf{P}^\mu + \mathbf{1}_{N\times N}]^{-1} \tag{4}$$

Substituting (4) into (1) reaches at (2). We will use (2) for numerical comparison of different policies in the following sections.

In this study we consider Poisson arrival process as the incoming traffic pattern to the transmitter side MAC buffer. In this case, it can be proved [19] that the MDP applied for system design in Section V results in a completely ergodic process with any *Markov deterministic* (MD) policy. Hence, according to [18], in the following discussion we focus on finding the optimal policy for the MDP without differentiating initial states.

There are two commonly used methods for finding the optimal policy of a MDP, i.e., *value iteration* and *policy iteration* [17], [18]. Since the convergent policy of value iteration method may not be precisely constructed in a theoretical manner[2], for MDP that continues for large, or infinite number of state transitions, which is the case discussed in this study, policy iteration method is more appropriate. Therefore, we apply the policy iteration method introduced in [18] for our design. This method can be summarized as the following steps:

1) Initialize the process gain g and a set of relative value function $\{v_1, v_2, \cdots, v_N\}$, which represents the differential values of each state $s \in \mathcal{S}$ with respect to v_N, as zero.

[1]In this study we focus on the application of Markovian deterministic policies [17]. In fact, for most Markov decision problems solved by history dependent policies, there exists a Markovian policy solving the same problem [17, p. 134]. Policies discussed in this paper are assumed stationary [17].

[2]Value iteration method is appropriate for finite horizon MDPs [18]. However, it is given in [17, p. 366] that the optimal policy obtained by value iteration when the process is $n \to \infty$ transitions away from termination, converges to a certain optimal policy, with a tolerated small deviation $\varepsilon > 0$ for finite value of n. Therefore, in this sense, value iteration method is also applicable for infinite horizon MDPs to find the ε-optimal policy.

2) *policy improvement routine*: Choose policy μ by finding action $a(s)$, for each state $s \in \mathcal{S}$, such that $a(s) := \arg\max_{a\in\mathcal{A}(s)} \left[q_s^a + \sum_{\omega\in\mathcal{S}} \left(p_{s;\omega}^a \times v_\omega \right) \right]$, where $\mathcal{A}(s)$ denotes the action space of state s.
3) *value determination operation*: Using $\mathbf{q}^\mu$ and $\mathbf{P}^\mu$ decided in step 2), solve g and v_s, $(s \in \mathcal{S}, s \neq N)$ through N linear equations, according to

$$g + v_s = q_s^\mu + \sum_{\omega\in\mathcal{S}} \left(p_{s;\omega}^\mu \times v_\omega \right) \quad (s = 1, 2, \cdots, N-1) \tag{5}$$

by setting $v_N = 0$.
4) If the new value of g obtained in step 3) is greater than its current record, update the record of g, μ and $\{v_1, v_2, \cdots, v_{N-1}\}$ and go to step 2). Otherwise, stop.
5) When selecting an action $a(s)$ for a new policy in step 2), if the corresponding action existing in the old policy is one of several equivalently good actions, retain the action choice in the old policy. This helps the final solution converge to a certain policy.
6) Once the iteration stops, the latest record of policy μ is the optimal policy[3].

Proposition 2: For a given policy μ decided in the *policy improvement routine*, the N linear equations entailed in the *value determination operation* can be solved as:

$$\mathbf{\Lambda} = (\mathbf{A} - \mathbf{B})^{-1} \times [\mathbf{q}^\mu]^T \tag{6}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Lambda}$ are defined, respectively as:

$$\mathbf{A} = \left[\begin{array}{c|c} 1 & \\ \vdots & \mathbf{I}_{N-1} \\ 1 & \\ \hline 1 & 0\cdots 0 \end{array} \right], \quad \mathbf{B} = \left[\begin{array}{c|c} 0 & \\ \vdots & \tilde{\mathbf{P}} \\ 0 & \end{array} \right], \quad \mathbf{\Lambda} = \left[\begin{array}{c} g \\ \tilde{\mathbf{v}} \end{array} \right]$$

with $\mathbf{P}^\mu = \left[\begin{array}{c|c} \tilde{\mathbf{P}} & \mathbf{P}_N \end{array} \right]$ and

$$\mathbf{P}_N = \left[p_{1;N}^\mu, p_{2;N}^\mu, \cdots, p_{N;N}^\mu \right]^T, \quad \tilde{\mathbf{v}} = [v_1, v_2, \cdots, v_{N-1}]^T$$

Proof: Setting $v_N = 0$, to solve $\tilde{\mathbf{v}}$ and g, we present the state transition probability matrix for given policy μ as $\mathbf{P}^\mu = \left[\begin{array}{c|c} \tilde{\mathbf{P}} & \mathbf{P}_N \end{array} \right]$. The N linear equations illustrated by (5) can be organized in matrix form as

$$\mathbf{A} \times \mathbf{\Lambda} = [\mathbf{q}^\mu]^T + \tilde{\mathbf{P}} \times \tilde{\mathbf{v}} = [\mathbf{q}^\mu]^T + \mathbf{B} \times \mathbf{\Lambda} \tag{7}$$

Again considering the nonsingularity of any full-ranked square matrix, with simplifications, the solution of g and $\{v_1, v_2, \cdots, v_{N-1}\}$ is obtained as the corresponding element of the $\mathbf{\Lambda}$ vector given in (6).

[3]The value of g for the optimal policy produced here can be obtained either through the policy iteration cycle or by (2). For any arbitrary policy, however, (2) is more appropriate.

V. Policy Domain Performance Optimization

The transmission mode selection approach mentioned in Section IV-A now can be translated into the MDP based optimization problem. In this case, the set of available transmission modes for each system state $s \in \mathcal{S}$ constitutes the action space of that state, i.e., $\mathcal{A}(s)$. Let us consider Poisson arrival process, which is given as:

$$p\{A(\tau) = i\} = \frac{e^{-\lambda\tau}(\lambda\tau)^i}{i!} \quad i = 0, 1, \cdots \tag{8}$$

where $A(\tau) = i$ denotes the event that i arrivals occur within the time duration of τ and λ is the average arrival rate in *packets/sec*. The system state can be completely characterized by the state pair of $s(k, q)$, where k and q denote the channel state index and MAC layer queue length in *packets* for a given state, respectively.

A. State Transition Probabilities and Rewards

In order to apply the MDP theory and thereby to find the optimal policy for transmission mode selection, the following items should be defined:

1) state transition probability matrix $\mathcal{T}$: The $\mathcal{T}$ matrix is three-dimensional, with the ath "slice" $\mathbf{\Upsilon}_a$ being a square matrix corresponding to action a as shown in (9), where B is the transmitter side MAC buffer size in *packets*. Therefore, the particular state transition probability matrix for any given policy μ, i.e., $\mathbf{P}^\mu$, with elements $p_{(k,q);(k',q')}$ is constructed by taking the corresponding row for state $s_{(k,q)}$ from the square matrix $\mathbf{\Upsilon}_{a^\mu(s_{(k,q)})}$, where $a^\mu\left(s_{(k,q)}\right)$ is the action defined by policy μ for state $s_{(k,q)}$.

The value of $p_{(k,q);(k',q')}^a$ in $\mathcal{T}$ is defined according to $\{a, k, q, k', q'\}$ as: For any $a \in \mathcal{A}\left(s_{(k,q)}\right)$,
1) if $| k - k' | < 2$ and $(q - \min(q, \varphi_{\max}^a) \le q' < B)$, then

$$p_{(k,q);(k',q')}^a = p\{A(T_f) = q' - [q - \min(q, \varphi_{\max}^a)]\} \times p_{k;k'} \tag{10}$$

2) if $| k - k' | < 2$ and $(q - \min(q, \varphi_{\max}^a) \le q' = B)$, then

$$p_{(k,q);(k',q')}^a = \sum_{j=B-[q-\min(q,\varphi_{\max}^a)]}^{\infty} p\{A(T_f) = j\} \times p_{k;k'} \tag{11}$$

where $\varphi_{\max}^a$ denotes the maximum number of packets that can be transmitted in T_f when transmission mode $a \in \mathcal{A}\left(s_{(k,q)}\right)$ is applied. For any other sets of $\{a, k, q, k', q'\}$, $p_{(k,q);(k',q')}^a = 0$.

These definitions are based on the following facts: 1) channel state transitions are only allowed for neighboring states as given in Section III; 2) at most $\min(q, \varphi_{\max}^a)$ packets can be serviced in any time frame; 3) packet arrival process can be viewed as independent from channel fading dynamic. The channel state transition probability $p_{k;k'}$ is given in [12],

$$\mathbf{\Upsilon}_a = \begin{bmatrix} p_{(1,0);(1,0)}^a & \cdots & p_{(1,0);(1,B)}^a & p_{(1,0);(2,0)}^a & \cdots & p_{(1,0);(K,B)}^a \\ p_{(1,1);(1,0)}^a & \cdots & p_{(1,1);(1,B)}^a & \cdots & \cdots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ p_{(K,B);(1,0)}^a & \cdots & \cdots & p_{(K,B);(2,0)}^a & \cdots & p_{(K,B);(K,B)}^a \end{bmatrix} \tag{9}$$

while $\mathcal{A}\left(s_{(k,q)}\right) := (mode\,0,\,1,\,\cdots,\,n)$ with mode n being the most conservative transmission mode (lower rate and less error) that can serve q packets in T_f.

2) state transition reward matrix $\mathcal{R}$*:* Each element of $\mathcal{R}$ specifies the reward value associated with the state transition defined by the corresponding transition probability in $\mathcal{T}$. We define the value of "reward" as the expected number of packets that will be correctly received at the receiver's MAC, resulted from corresponding state transition of the system. With this definition, the elements of $\mathcal{R}$ is defined for $\{a, k, q, k', q'\}$ as:

$$\begin{matrix} r^a_{(k,q);(k',q')} \\ \forall\, s_{(k',q')} \in \mathcal{S} \end{matrix} = \begin{cases} \min(q, \varphi^a_{\max}) \times \left[1 - P^a_p(k)\right] \\ a \in (mode\,1,\,\cdots,\,n)\,,\; p^a_{(k,q);(k',q')} \neq 0, \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{12}$$

where $P^a_p(k)$ denotes the average PER when the channel is in state k and transmission mode $a \in \mathcal{A}\left(s_{(k,q)}\right)$ is applied.

Having the $\mathcal{T}$ and $\mathcal{R}$ matrixes ready, the optimal policy for transmission mode selection depicted in Section IV-A can be solved by the policy iteration method mentioned in Section IV-B.

B. Analytical Model

In this subsection we develop a general analytical framework that relates different policy domain design schemes with various QoS performance metrics.

Supposing any policy μ is applied to dynamically select the transmission mode, in steady state, the resultant Markov process is completely characterized by the state transition probability matrix $\mathbf{P}^\mu$, which is a subset of $\mathcal{T}$ as interpreted in Section V-A1. Utilizing (4) and (2), the steady state probability distribution vector $\boldsymbol{\pi}$[4] and the process gain g can be solved accordingly. Based on these quantities, multiple QoS performance metrics can be derived.

1) Average Throughput: The average throughput η, in *packets/sec*, can be obtained from the process gain in (2), due to appropriate definition of "reward", as: $\eta = g/T_f$.

2) Average Packet Loss Rate: Despite the source of packet loss, i.e., transmission error or buffer overflow, lost packets are just the complementary portion of correctly delivered packets and backlogged packets in the queue, with respect to the total number of arrivals. Hence, the average packet loss rate, denoted by ξ, is given as:

$$\xi = \lim_{t\to\infty} \frac{\lambda t - \frac{g}{T_f}t - L(t)}{\lambda t} = 1 - \frac{g}{\lambda T_f} \tag{13}$$

where $L(t)$ is the instant queue length in *packets* at time t and $L(t) \leq B$.

3) Average Queuing Delay: The queuing delay here is measured in the number of delayed time frames. By Little's theorem [20], the average queuing delay D is given as: $D = \frac{L}{\lambda_q T_f}$, where L is the average queue length and λ_q denotes the corresponding average arrival rate of the enqueued packet stream. Namely, the portion of dropped packets due to buffer overflow is not included. Here, the value of L can be easily obtained from the marginal distribution of the queue size as: $L = \sum_{j=0}^{B}\left(j \times \sum_{i=1}^{K} \pi_{(i,j)}\right)$, where $\pi_{(i,j)}$ denotes the steady state probability of system state $s_{(i,j)} \in \mathcal{S}$. To evaluate λ_q, we noticed that in steady state it is equal to the average departure rate from the queue. Therefore, λ_q can be obtained equivalently by finding the average number of packets dequeued from the sending MAC per time frame, divided by T_f. Particularly, it is given as:

$$\lambda_q = \frac{1}{T_f}\sum_{i=1}^{K}\sum_{j=0}^{B}\left[\min\left(j,\, \varphi_{\max}^{a(s_{(i,j)})}\right) \times \pi_{(i,j)}\right] \tag{14}$$

4) Average PER: Once λ_q is given, the average PER over the channel, i.e., P_p, can be obtained readily as:

$$P_p = \lim_{t\to\infty} \frac{\lambda_q t - \frac{g}{T_f}t}{\lambda_q t} = 1 - \frac{g}{\lambda_q T_f} \tag{15}$$

Equation (15) utilizes the fact that the total number of dequeued packets from the sending MAC is the sum of the total number of correctly delivered packets and the total number of erroneously transmitted packets.

5) Average Packet Dropping Probability: Finally, packets being dropped upon arrival just constitute the complementary portion of enqueued packet stream, with respect to the total number of arrivals. Therefore, the average packet dropping probability, denoted as P_d, is given as:

$$P_d = \lim_{t\to\infty} \frac{\lambda t - \lambda_q t}{\lambda t} = 1 - \frac{\lambda_q}{\lambda} \tag{16}$$

The above analytical model can be used to compare the performance of different policy designs. Below we first validate this model by running Monte Carlo simulations.

C. Model Verification

In the simulation we apply two different policies, i.e., μ^*, the optimal policy that minimizes ξ and is obtained by the policy iteration method introduced in Section IV-B, and μ_0, the fixed policy defined in [3]. In stead of optimizing the policy design for each CPM, μ_0 constantly requires the transmission mode being switched from one to another, with a fixed (transmission)mode-(channel)state mapping, for different CPMs. This imperfectness in fact encouraged the policy domain optimization proposed in this section.

1) Simulation Setup: In order to closely compare the performance of μ_0 and μ^*, we apply the CPMs introduced in [3], though other CPMs are also applicable. The CPMs proposed in [3] utilize a reference parameter P_0, i.e., the target average packet error rate over the channel, to control the channel partition. Hence, each value of P_0 defines a particular CPM. To maintain a clear concept, below we refer to the optimal policy with respect to the CPM defined by particular value of P_0 as $\mu^*(P_0)$. Note that μ_0 is invariant for any CPM. The relevant parameters used for simulation and corresponding analytical computation are listed in Table I.

The average PER by different transmission modes of TM2 with presence of *additive white Gaussian noise* (AWGN), are approximated as [3, eq. (3)]:

$$PER_n(\gamma) \approx \begin{cases} 1, & 0 < \gamma < \gamma_{pn} \\ a_n \exp\left(-g_n\gamma\right), & \gamma \geq \gamma_{pn} \end{cases} \tag{17}$$

[4]Below we omit the superscript μ for notational simplicity.

TABLE I
SIMULATION PARAMETERS

ρ	15 dB
f_m	10 Hz
T_f	1 ms
λ	1000 $packet/second$ (1 $packet/frame$)
B	15 $packets$
L_p	1080 $bits$
symbol rate	2.16$MBaud$ (2160 $symbol/frame$, or $b = 2$ as in [3])

where $n = 1, \cdots, 5$ is the transmission mode index and γ is the received SNR per symbol. The corresponding value of fitting parameters a_n, g_n and r_{pn} for packet length $L_p = 1080\, bits$ are given by Table II in [3].

2) Results and Discussions: The values of P_p and P_d obtained via simulation and analysis are illustrated in Fig. 2 and Fig. 3, respectively. It is recognized that with the fixed policy μ_0, both of P_p and P_d exhibit monotonous behavior when P_0 is loosed. This implies a balanced point where the joint effect of P_p and P_d, i.e., the MAC layer throughput or overall packet loss, is optimized. This value of P_0 (i.e., between 10^{-3} and 10^{-2}) is clearly shown in Fig. 4 and Fig. 5, where the MAC layer throughput and average packet loss rate performances are visualized. In contrast to the passive response of μ_0 to the variation of CPM[5], the policy domain design for $\mu^*(P_0)$ proactively enforces more aggressive transmission (higher rate and more errors) when the air budget is tight (smaller values of P_0), and more conservative transmission when the buffer crisis is comparatively non-prominent (larger values of P_0). Therefore, when the applied CPM varies, the value of P_p and P_d covary actively, while consistently to minimize the joint effect of these two, as also shown in Fig. 2–Fig. 5. Finally, the average packet queuing delay compared in Fig. 6 reveals that when P_0 is loosed, differentiated from $\mu^*(P_0)$, μ_0 improvidently permits packets to pass through the MAC buffer faster and faster without considering the joint performance at both PHY and MAC layers.

From Fig. 2 to Fig. 6 it is seen that the simulated curves well match the corresponding analytical computations, in terms of various performance metrics. In the later sections we will further investigate optimization design methods only using the validated analytical model.

VI. TWO-DIMENSIONAL PERFORMANCE OPTIMIZATION

Up to now we have mainly focused on the discussion of optimizing the policy for transmission mode selection, i.e., policy domain optimization, based on particular CPM. However, the potential improvement by searching for the best CPM, given the optimal policy for each partition method, i.e., channel domain optimization, has not been investigated. Though applying fixed policy μ_0 for each CPM, the design approach proposed in [3] aims to explore performance optimization from this perspective. In the following discussion we refer to this approach as *one dimensional* (*1-D*) optimization scheme. The other approach, which we refer to as *two dimensional* (*2-D*) optimization scheme, is the one that incorporates both

[5]Note that for μ_0 the average packet error rate over the channel is constantly P_0, while for $\mu^*(P_0)$ this value is contingent on particular policy design.

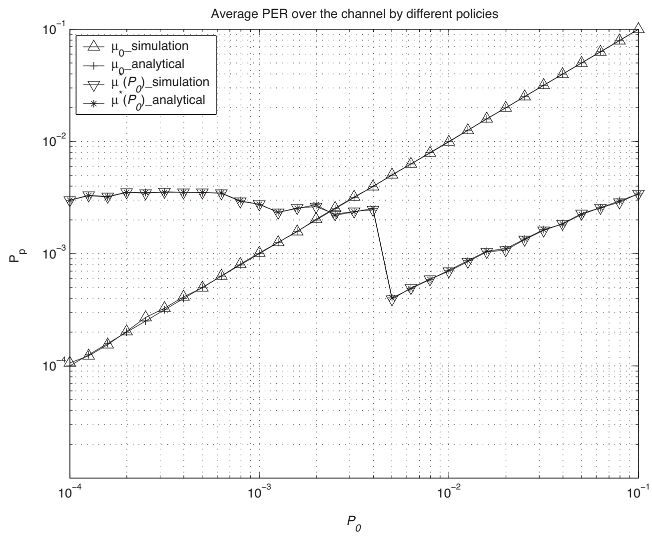

Fig. 2. Average packet error rate by different policies.

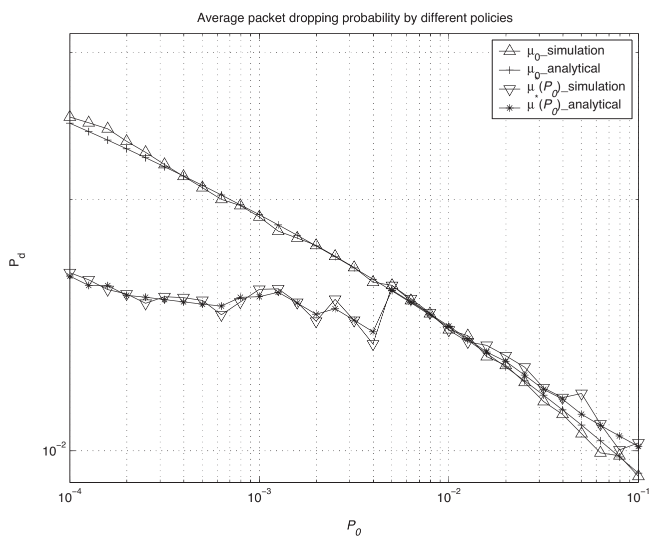

Fig. 3. Average packet dropping probability by different policies.

policy domain optimization and channel domain improvement to achieve the best solution.

Apparently, finding the optimal CPM is equivalently to determine the optimal value of P_0 that offers the minimum attainable ξ, using the corresponding optimal policy. As given in [3], this optimal value of P_0, i.e., P_0^{opt}, can be found by numerically comparing the value of $\xi(P_0)$ for all possible choices of P_0, i.e., $P_0^{opt} := \arg\min_{P_0 \in \mathcal{P}} \xi(P_0)$, where $\mathcal{P}$ denotes the set of possible P_0 choices.

In Fig. 7, as example, the performance of *1-D* and *2-D* optimization schemes are presented by varying the value of λ, to show the advancement of *2-D* optimization design. Except varying λ, other remaining parameters are set by Table I. From Fig. 7 we can conclude that 1) policy domain optimization is indeed contributive, as the curve for *2-D* optimization is constantly below the corresponding curve for *1-D* optimization; 2) channel domain optimization makes appreciable improvement, as each curve fluctuates when the CPM varies;

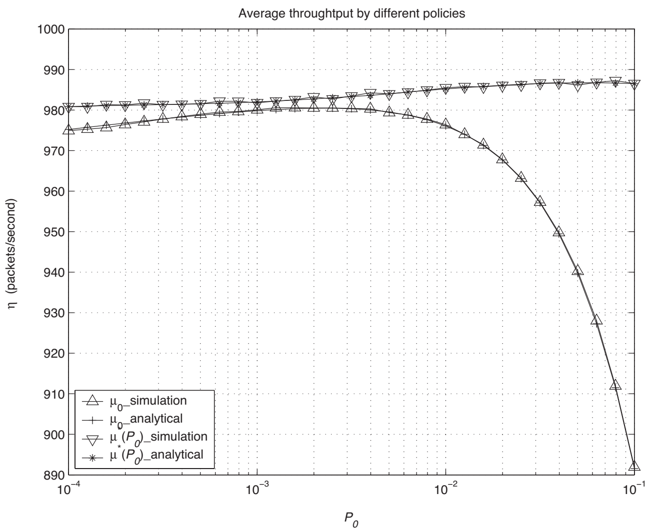

Fig. 4. Average throughput by different policies

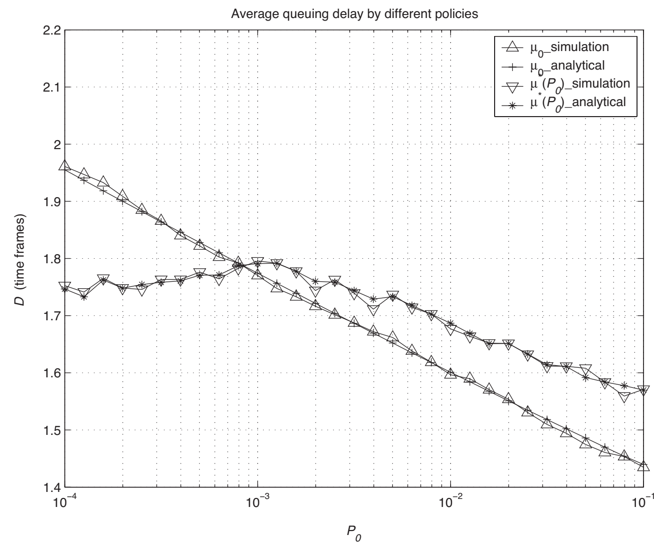

Fig. 6. Average packet queuing delay by different policies

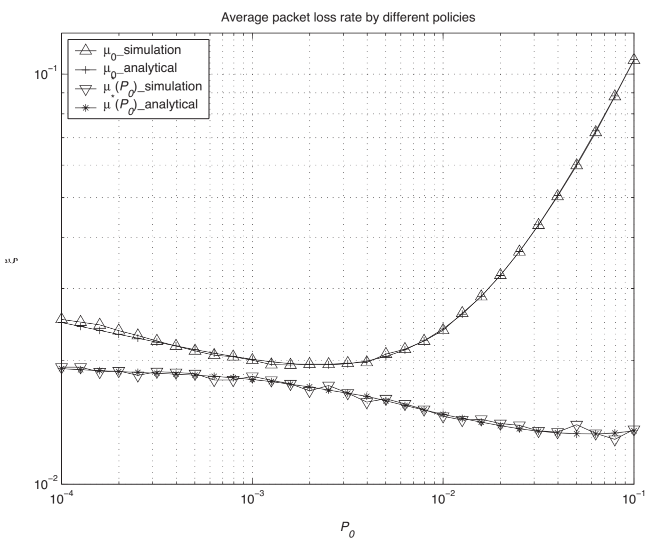

Fig. 5. Average packet loss rate by different policies

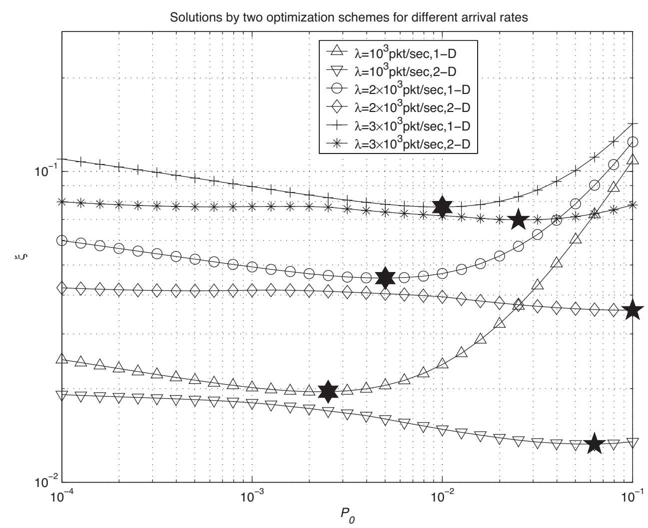

Fig. 7. Average packet loss rate by *1-D* and *2-D* optimization schemes with different arrival rate (solid hexagrams indicate the minimum attainable values for ξ by *1-D* optimization, solid pentagrams indicate such values by *2-D* optimization).

3) *2-D* optimization always offers lower minimum attainable average packet loss rate, compared with *1-D* optimization, which confirms the initial motivation of this study.

VII. Complexity-aware 2-D Suboptimal Design

In the previous sections we have not touched the complexity issue of the proposed optimal solutions. The advantage of *1-D* design over *2-D* design lies in the MAC-blind feature of *1-D* design for simple implementation, especially when the transmitter side buffer size is large. To enhance the implementability of the proposed *2-D* design solution, in this section we suggest a suboptimal method that achieves scalable complexity-optimality tradeoff.

A. Existing Complexity-aware Suboptimal Solutions

Complexity-aware scheduling design for wireless communication involving fading channel has gained research attention for a while. Notable research on this topic include [5] and [8]. In [5] the authors devised a *log-scheduling* algorithm, where the number of packets to be transmitted in time frame n is decided, based on the conditional average SNR of the current channel state $\tilde{h}_n$ and buffer occupancy $\mathbf{B}_n$, as:

$$\mathbf{U}_n = \max\left(\min\left(\mathbf{U}_{atbest}, \left\lfloor \log\left(\tau \mathbf{B}_n \left(\tilde{h}_n\right)^{\kappa}\right)\right\rfloor\right), \mathbf{U}_{atleast}\right) \quad (18)$$

where $\mathbf{U}_{atbest}$ enforces the fact that no more packets than the current queue backlog are serviced, and $\mathbf{U}_{atleast}$ ensures that adequate buffer space is vacated in frame n to accommodate incoming packets in frame $n+1$. By appropriately setting the value of τ and κ in (18), the *log-scheduling* algorithm is tested as capable of approximating the performance of theoretical optimal solution. In [8], a simple transmission scheduling method is introduced. In this paper the number of packets to be serviced in time frame n when the queue length is x_n, is determined as $r_n = \min(x_n, r_a)$ if $\gamma^{(k+1)} \geq \gamma_a$, and

$r_n = 0$ otherwise, where $\gamma^{(k+1)}$ is the upper bound SNR value of the current FSMC channel state $s^{(k)}$, r_a and γ_a are predesigned parameters that enable the suboptimal scheduling algorithm to perform similarly with the DP based optimal solution. Moreover, for a given parameter setting of γ_a and r_a, in [8] various QoS performance metrics are also derived.

Though the above suboptimal schemes can considerably alleviate the operational complexity in the implementation space, the design space methodology that can effectively locate the respective suboptimal solution is not well defined. For example, the value of τ and κ in [5], or the value of γ_a and r_a in [8], that offer close approximation of the corresponding optimal solution, are only determined through numerical methods. In this section, we introduce a suboptimal design approach that 1) in the implementation space, offers scalable complexity-optimality tradeoff for different hardware configurations; 2) in the design space, specifies tractable operation to produce the appropriate suboptimal solution.

B. 2-D Suboptimal Design

For practical implementation, in general the channel can be appropriately partitioned into a small number of states. However, when the transmitter side buffer size is large, the sending AMC module has to perform complex table-lookup operation, on frame-by-frame basis, to find the predesigned transmission mode for that instantaneous system state. To alleviate this operational complexity, we segment the packet-level queue states into consecutive non-overlapping sections as $\Omega := (\Phi_1, \Phi_2, \cdots, \Phi_h)$ $(1 \leq h \leq B+1)$. Within each section, the queue states are mutually indistinctive. Hence, the frame-by-frame table-lookup operation is only performed over $K \times h$ entries, with h being a scale parameter for different hardware configurations. Formally, we have constructed an alternative representation of the system states as $\mathcal{V} := \nu_{(k,\Phi)}$ $(1 \leq k \leq K,\ \Phi \in \Omega)$, based on which a suboptimal scheduling policy is to be determined in the design space. Let $\Omega^{(C,h)}$ denote the case where $C = B+1$ queue states (B nonempty states and one empty state) are segmented into h sections. By varying the value of h, a series of suboptimal solutions can be achieved. Particularly, when $h = C$, this *2-D* suboptimal scheme repeats the *2-D* optimal solution discussed in section VI; and when $h = 1$, it exhibits the same implementational complexity as the MAC-blind *1-D* optimization design applied in [3], however with better optimality that will be shown later.

In the design space, to determine such suboptimal policy for a given $\Omega^{(C,h)}$, we apply an *extended policy iteration* (EPI) method in the policy domain optimization phase discussed in Section V. Specifically, the new design target can be formulated as to determine the suboptimal policy μ^{subopt}, such that $\mu^{subopt} := \arg\min_{\mu \in \mathcal{U}} \xi(\mu)$, subject to $a^{\mu}\left(s_{(k,q\in\Phi)}\right) = a^{\mu}\left(\nu_{(k,\Phi)}\right)$, where $\mathcal{U} := \mathcal{A}(\nu_{(k,\Phi)}) \times \mathcal{V}$ and $\mathcal{A}(\nu_{(k,\Phi)})$ consists of transmission modes $0, 1, \cdots, n$ with n being the most conservative transmission mode that can carry the maximal queue backlog in Φ during T_f. To comply with this new design target, we modify the *policy improvement routine* step given in Section IV-B as follows:

Policy improvement routine: for each state $\nu_{(k,\Phi)} \in \mathcal{V}$, choose action $a\left(\nu_{(k,\Phi)}\right) \in \mathcal{A}\left(\nu_{(k,\Phi)}\right)$ such that $a\left(\nu_{(k,\Phi)}\right) :=$

$$\arg\max_{a \in \mathcal{A}\left(\nu_{(k,\Phi)}\right)} \mathbb{E}\left[q^a_{s_{(k,q\in\Phi)}} + \sum_{(k',q')\in\mathcal{S}} \left(p^a_{(k,q\in\Phi);(k',q')} \times v_{(k',q')}\right)\right] \tag{19}$$

The expected value $\mathbb{E}(\cdot)$ in (19) is computed as the weighted sum of the operand by assigning a weight $w_{s_{(k,q)}}$, for each state $s_{(k,q)} \in \nu_{(k,\Phi)}$, as: $w_{s_{(k,q)}} = \frac{\pi_{s_{(k,q)}}}{\sum_{s_{(k,q')}\in\nu_{(k,\Phi)}} \pi_{s_{(k,q')}}}$. The value of $\pi_{s_{(k,q)}}$ is updated along with the relative values $v_{s_{(k,q)}}$ in the *value determination operation* step introduced in Section IV-B and according to (4). In the initialization step these weights can be simply set for each state $s_{(k,q)} \in \nu_{(k,\Phi)}$ as $1/N(\Phi)$, where $N(\Phi)$ is the number of queue states included in section Φ.

Remark 1: The above EPI method performs well for the first few iterations where the process gain increases very aggressively, as by the original policy iteration method. However, when the current policy is closely converging to the desired suboptimal policy where the process gain increases very slightly, it is possible (but not always) and as we have observed, that the two policies obtained by consecutive iterations may quibble between two nearly-equivalently performed alternatives (e.g., between $g = 5.9582$ and $g = 5.9621$). This is due to the deviation introduced by using the weights calculated from the old policy to generate the new policy[6]. Nevertheless, the iteration can be easily stopped in either cases, i.e., quibbling or non-quibbling, as: if the same policy appears twice or more in the most recent three iterations, stop the algorithm and select the best policy obtained within these three iteration cycles.

Remark 2: It is helpful to recognize that the design space complexity for locating an appropriate suboptimal solution, either by the numerical methods used in [5] and [8] or the EPI method introduced here, has not been remarkably depreciated. However, the issue of design space complexity is usually secondary, since when constructing a practical system we always prefer to transfer as much as possible complexity from the online implementation into the offline design.

C. Numerical Results and Discussions

In this subsection, we use the analytical model developed in Section V-B to visualize the scalable complexity-optimality tradeoff offered by the proposed *2-D* suboptimal solutions. In general, the queue state segmentation can follow any distribution function $\chi\left(\Omega^{(C,h)}\right)$. Let us consider the simple *identical segmentation* (IS) case, denoted as $\chi^{IS}\left(\Omega^{(C,h)}\right)$, where the C queue states are segmented into h sections, with each section equivalently includes C/h queue states (without losing generality, we can assume an integer value). The transmitter side MAC buffer size is now set to $199\,packets$, i.e., $C = 200$

[6]In strict sense, this deviation also affects the first few iterations. However, at that stage there are ample of policy alternatives that can largely differentiate the relative values than the current policy does. It is thus almost impossible, or with small probability, to determine a wrong policy when the process gain is able to increase aggressively. More strictly speaking, even this quibbling occurs when the process gain is unexpectedly small, the iteration can be restarted by initializing the weights with different values.

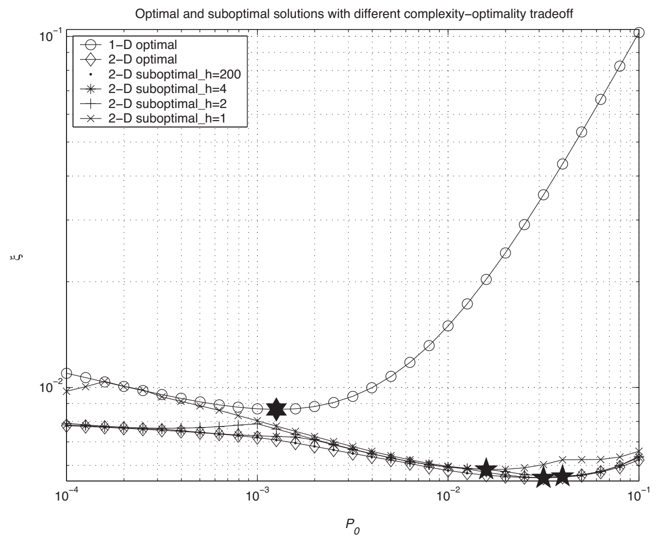

Fig. 8. Average packet loss rate by *1-D* and *2-D* optimal and suboptimal solutions with different complexity-optimality tradeoff (solid hexagram indicates the minimum attainable value of ξ by *1-D* optimization, solid pentagrams indicate such values by *2-D* suboptimization for the case of $\chi^{IS}\left(\Omega^{(200,h)}\right)$).

queue states. In order to induce better queuing dynamic thereby to eliminate the virtual impossibility of large buffer occupancy, we also increase the value of λ and the physical layer transmission symbol rate to $6 \times 10^3\, packets/second$ and $10.8M\, Baud$ (i.e., $b = 10$ in [3][7]), respectively. With other parameters still set by Table I, Fig. 8 illustrates the performance of *2-D* suboptimal solutions for $\chi^{IS}\left(\Omega^{(200,h)}\right)$ with $h = 1, 2, 4, 200$, as well as the optimal solutions obtained by *1-D* and *2-D* optimization schemes discussed in Section VI. As expected, the *2-D* suboptimal solutions are capable of offering better optimality when the value of h increases, in terms of minimizing ξ. Also, when $h = C = 200$, the *2-D* suboptimal curve overlaps with the *2-D* optimal curve depicted in Section VI. It is interesting to notice that with the same implementational complexity, the *2-D* suboptimal curve for $h = 1$ offers much lower minimum attainable average packet loss rate than the *1-D* optimal curve applied in [3]. The reason lies in the contribution of policy domain optimization performed by *2-D* solutions. Namely, the policy designed for μ^{subopt} is allowed to vary for different CPMs, instead of being fixed as μ_0.

VIII. Future Extensions

1) Automatic Repeat Request (ARQ): One possible extension of this work is to embed link layer ARQ into the proposed *2-D* optimization design. This mainly requires some modification in the $\mathcal{T}$ matrix detailed in Section V-A1, considering the fact that erroneously transmitted packets in time frame n remain in the queue for retransmission at the next decision epoch nT_f. Since MAC layer throughput is not lost over the channel, in this case, the optimization target can be uniquely defined as either MAC layer throughput or the buffer overflow probability. Moreover, Long has studied on the ARQ-enhanced performance of Liu's model [9] and developed a more general analytical model for multiuser ARQ system with any given scheduling scheme [21].

2) Network Application: Another possible extension of this work is to have the per-user decision made in a network context, i.e., with $n > 1$ transmitter-receiver pairs, where selfish PHY layer behavior may degrade the network level MAC layer throughput due to anarchical inter-user interference. Particularly, we are considering Telatar's processor-sharing model [14], where jobs are offered by the transmitters and the receivers collectively act as single virtual server distributing information-theoretic service rate among jobs. The network level MAC layer capacity (maximal possible throughput) is a random process and can be defined as $\sum_{1\le i\le n} c_i(t)$, where $c_i(t)$ is the maximal possible MAC layer data rate distributed to user i based on its instantaneous channel state and MAC buffer occupancy at time t, taking other $(n-1)P$ transmission power as noise (see (1) in [14]). When user i selects transmission mode at a decision epoch, as associated "reward", it receives corresponding expected MAC layer throughput if any mode $x \neq 0$ is selected, and $\sum_{1\le j\le n, j\neq i} c'_j(t) - \sum_{1\le j\le n} c_j(t)$ if mode 0 is selected (may be negative), where each $c'_j(t)$ is computed with $(n-2)P$ transmission power being taken as inter-user noise. With the policy domain optimization discussed in Section V, each user can select its optimal per-(time)frame transmission mode for maximizing the long-term network throughput. Here we should assume some overhead for user interaction to obtain network wide knowledge. This extension is under research and findings will be reported later.

IX. Conclusion

In this study we have proposed a two-dimensional cross-layer performance optimization scheme for packet transmission withstanding wireless fading channel. Without run-time power adjustment, this technique applies concurrently both policy domain optimization for AMC rate adaptation and channel domain improvement to form the FSMC model. Alongside the optimal solution, a suboptimization framework was proposed to obtain scalable complexity-optimality tradeoff for practical implementations.

Acknowledgment

The authors would like to thank the anonymous reviewers for their insightful comments on improving the quality of the paper. They also thank Prof. S. L. Primak in Department of Electrical and Computer Engineering, The University of Western Ontario, for early discussion on FSMC modeling.

References

[1] X. Zhang, J. Tang, H.-H. Chen, S. Ci, and M. Guizani, "Cross-layer-based modeling for quality of service guarantees in mobile wireless networks," *IEEE Commun. Mag.*, vol. 44, no. 1, pp. 100–106, Jan. 2006.

[2] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer design for wireless networks," *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 74–80, Oct. 2003.

[3] Q. Liu, S. Zhou, and G. B. Giannakis, "Queuing with adaptive modulation and coding over wireless links cross-layer analysis and design," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 1142–1153, May 2005.

[7] In [3] eq. (16), b is defined as $b = (N_s - N_c)/N_b$. In our context, this implies that given the value of b, the physical layer transmission symbol rate can be computed as $b \times L_p/T_f\ Baud$.

[4] R. A. Berry and R. G. Gallager, "Communication over fading channels with delay constraints," *IEEE Trans. Inform. Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.

[5] A. K. Karmokar, D. V. Djonin, and V. K. Bhargava, "Optimal and suboptimal packet scheduling over correlated time varying flat fading channels," *IEEE Trans. Wireless Commun.*, vol. 5, no. 2, pp. 446–456, Feb. 2006.

[6] I. Bettesh and S. Shamai, "Optimal power and rate control for fading channels," *IEEE VTC*, vol. 2, pp. 1063–1067, May 2001.

[7] M. Goyal, A. Kumar, and V. Sharma, "Power constrained and delay optimal policies for scheduling transmission over a fading channel," *IEEE INFOCOM*, vol. 1, pp. 311–320, Mar. 2003.

[8] H. Wang and N. B. Mandayam, "A simple packet-transmission scheme for wireless data over fading channels," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1055–1059, July 2004.

[9] L. B. Le, E. Hossain, and A. S. Alfa, "Radio link level performance evaluation in wireless networks using multi-rate transmission with arq-based error control," *IEEE Trans. Wireless Commun.*, vol. 5, no. 10, pp. 2647–2653, Oct. 2006.

[10] L. Lin, R. D. Yates, and P. Spasojević, "Adaptive transmission with finite code rates," *IEEE Trans. Inform. Theory*, vol. 52, no. 5, pp. 1847–1860, May 2006.

[11] A. Fu, E. Modiano, and J. Tsitsiklis, "Optimal energy allocation for delay-constrained data transmission over a time-varying channel," *IEEE INFOCOM*, vol. 2, pp. 1095–1105, Mar. 2003.

[12] H. S. Wang and N. Moayeri, "Finite-state markov channel–a useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, no. 1, pp. 163–171, Feb. 1995.

[13] Q. Zhang and S. A. Kassam, "Finite-state markov model for rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688–1692, Nov. 1999.

[14] I. E. Telatar and R. G. Gallager, "Combining queueing theory with information theory for multiaccess," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 963–969, Aug. 1995.

[15] M. K. Simon and M.-S. Alouini, *Digital Communication over Fading Channels: A Unified Approach to Performance Analysis*, J. G. Proakis, Ed. New York: John Wiley & Sons, 2000.

[16] A. J. Goldsmith and S.-G. Chua, "Variable-rate variable-power MQAM for fading channels," *IEEE Trans. Commun.*, vol. 45, no. 10, pp. 1218–1230, Oct. 1997.

[17] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: John Wiley & Sons, 1994.

[18] R. A. Howard, *Dynamic Programming and Markov Processes*. Cambridge: The M.I.T. Press, 1960.

[19] X. Bai and A. Shami. (2007, Aug.) On the ergodicity of wireless packet transmission with Markovian deterministic scheduling policies and memoryless traffic inputs. [Online]. Available: http://www.eng.uwo.ca/people/ashami/Papers/NetRep-2007-03.pdf

[20] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, N.J.: Prentice Hall, 1992.

[21] L. B. Le, E. Hossain, and A. S. Alfa, "Delay statistics and throughput performance for multi-rate wireless networks under multiuser diversity," *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3234–3243, Nov. 2006.

Xiaofeng Bai received the B.E. degree in electrical engineering from Shandong University, China, in 1997, and the M.E.Sc. degree in electrical and computer engineering from the University of Western Ontario, Canada, in 2005. He is currently working towards the Ph.D. degree in electrical and computer engineering at the University of Western Ontario. His research interests are in the area of quality-of-service in broadband access networks, cross-layer protocol design in wireless communication systems, network simulation and modeling.

Abdallah Shami received the B.E. degree in electrical and computer engineering from the Lebanese University, Beirut, Lebanon, in 1997, and the Ph.D. degree in electrical engineering from the Graduate School and University Center, City University of New York, New York, in September 2002. In September 2002, he joined the Department of Electrical Engineering at Lakehead University, ON, Canada, as an Assistant Professor. Since July 2004, he has been with the University of Western Ontario, London, ON, Canada, where he is currently an Assistant Professor in the Department of Electrical and Computer Engineering. His current research interests are in the area of wireless/optical networking, EPON, WIMAX, and WLANs. Dr. Shami received the Irving Hochberg Dissertation Fellowship Award and a Graduate Teaching Fellowship at the City University of New York.